The Vigenère Cipher

The Vigenère cipher is a classic encryption technique that dates back to the 16th century. It was invented by a French diplomat and cryptographer named Blaise de Vigenère, hence the name.

It is a polyalphabetic substitution cipher that uses a keyword to encrypt and decrypt messages. The key idea behind the Vigenère cipher is to use multiple Caesar ciphers based on the letters of the keyword.

Let's say we have a plaintext message P consisting of n letters. We also have a keyword K consisting of m letters, where m ≤ n.

To encrypt the message, we repeat the keyword until it matches the length of the plaintext. To encrypt each letter p_i in the plaintext, we find the corresponding letter k_j in the keyword K' where j is the index of p_i modulo m . Then, we shift the letter p_i by the index of k_j in the alphabet. Let E represent the encrypted message.

The mathematical representation of the encryption process is given by: E_i = (p_i + k_j) mod 26, where E_i is the i-th letter of the encrypted message and j = (i − 1) mod m.

To decrypt the message, we use a similar process. We use the method of index of coincidences (Friedman's method) to find the length of the keyword. We then use frequency analysis to find the keyword. We repeat the keyword until it matches the length of the ciphertext. To decrypt each letter E_i in the ciphertext, we find the corresponding letter k_j in the keyword K' (where j is the index of E_i modulo m). Then, we shift the letter E_i back by the index of k_j in the alphabet. Let D represent the decrypted message.

The mathematical representation of the decryption process is given by: D_i = (Ei – kj) mod 26, where D_i is the i-th letter of the decrypted message and j = (i − 1) mod m.

Note that in these equations, we use modular arithmetic with 26 since there are 26 letters in the English alphabet. Additionally, we assume a simple mapping where A is represented by 0, B by 1, and so on.

Flowchart:

Ask user to decrypt or encrypt

Ask user to enter string and keyword

Validate string and keyword:

- Use isalpha() and isblank()

Separate each letter to element in array

Use switch case to map letter to number